Classifying the Response Space of Questions: A Machine Learning Approach

Zulipiye Yusupujiang
Université Paris Cité, CNRS
Laboratoire de Linguistique Formelle
`zulipiye.yusupujiang@etu.u-paris.fr`

Alafate Abulimiti
INRIA, Paris, France
`alafate.abulimiti@inria.fr`

Jonathan Ginzburg
Université Paris Cité, CNRS
Laboratoire de Linguistique Formelle
`yonatan.ginzburg@u-paris.fr`

Abstract

The main goal of this work is to conduct a pilot study on the automatic classification of the response space of questions in English. We aim for a relatively fine-grained understanding of the learning problem of this response space; hence, we conducted classical machine learning studies to automatically identify different response classes based on carefully designed features. Moreover, we compared the results from feature-based classical machine learning algorithms to the classification results obtained from a large-scale pre-trained BERT language model. Experimental results show that the feature-based classical machine learning algorithms can achieve performance results which are close to the results obtained by BERT model on this novel task. The overall trend of the classification results for each response class are also similar in both models. Learnability trends similar to corpus-based studies presented in previous literatures emerge.

1 Introduction

Classifying the response space of questions plays an important role in the design of dialogue systems, particularly systems that can be easily adaptable across domains (Larsson and Berman, 2016). Łupkowski and Ginzburg (2013, 2016) offer an empirical and theoretical characterization of one significant component of the response space of questions, which is responding to a question with a question, which represents more than 20% of all responses to questions found in the British National Corpus (BNC) (Burnard, 2007). Based on a detailed corpus study on the British National Corpus and three other more genre-specific corpora (BEE (Rosé et al., 1999) and AmEx (Kowtko and Price, 1989)) and a sample from CHILDES (MacWhinney, 2000)), Łupkowski and Ginzburg (2013, 2016) provide 7 classes of question responses: `CR`: *clarification requests*, `DP`: *dependent questions*, `MOTIV`: *requests for underlying motivation*, `FORM`: *questions about the form of the expected answer*, `NO ANSW`: *questions raised with the aim of not answering the initial question*, `IND`: *questions providing a potential answer*, and `IGNORE`: *questions raised to ignore the initial question*.

Following the aforementioned research, Ginzburg et al. (2019, 2022) extend the classification of response space to cover all responses to questions. They provide a full response space taxonomy with 9 unique response classes of responses to questions and one OTHER class. They conduct cross-linguistic studies comparing English and Polish.

The main aim of the current work is to conduct a pilot study for automatic classification of response space of questions, based on the taxonomy proposed by Ginzburg et al. (2019, 2022). Such an approach lays a foundation for the automation of response space classification in designing dialogue systems.

This paper is structured as follows: In section 2, we discuss related work on classifying other types of utterances in dialogue. Section 3 contains a discussion of the taxonomy of responses to questions used in this study. In Section 4, we introduce the response space annotation process and labeled dataset. Section 5 presents the experiments on BERT language model and its results. We then introduce the specifically created feature sets, and discuss the results and learnability of different response classes from a classical machine learning algorithm in Section 6. In the last section, we offer some conclusions and discuss future work aimed at improving this study.

2 Related Work

Fernández et al. (2007) propose a taxonomy with 15 classes for Non-Sentential Utterances (NSU) in dialogue, based on a detailed corpus study on BNC. In addition, they also present several results from automatically classifying NSUs using some well-known machine learning techniques. For the machine learning approach, they use the majority class predictor, one-rule classifier, and also the J4.8 decision tree algorithm using the Weka Toolkit (Witten and Frank, 2002). Classification results from the algorithms above served as the baselines of their study. Three other machine learning systems were also used, SLIPPER (Cohen and Singer, 1999), TiMBL (Daelemans et al., 2003), and MaxEnt (Zhang, 2007), in order to conduct a more sophisticated experiment and get a reliable result. To train the machine learning algorithms, Fernández et al. (2007) used three types of feature sets which capture either the properties of NSUs, of the antecedent utterance, or the relations between NSUs and the antecedents. Their results show that machine learning algorithms benefit from utilizing the properties of the antecedent of NSUs and also the relationships between them.

Dragone and Lison (2015) propose an active learning approach to the classification of NSUs, by an extension of the work of Fernández et al. (2007). They extend the feature set from 9 features to a total of 32 features by extracting more features with the PCFG and Dependency Parser from the Standford CoreNLP API (Dragone and Lison, 2015). An active learning method is used to deal with the labelled data scarcity problem. The experimental results show a significant improvement on the classification task when comparing it to the baseline of Fernández et al. (2007). In this study, we use similar methods used to classify NSUs as discussed above.

Clarification requests (CRs) are also common in human dialogue. According to Purver et al. (2003a); Rodríguez and Schlangen (2004), CRs account for 3%-6% of human-human dialogue. CRs are also common in response space taxonomy (4.84% as shown in Table 2). Purver (2006) studies Clarification Requests in details and presented all major forms of CRs and analyzed their readings. He also offered a computational implementation of CRs within a prototype text-based dialogue system - CLARIE.

In addition, Cruz-Blandón et al. (2019) propose a semantic annotation scheme for questions and answers based on the contribution of content and discourse on them. They divided the questions into 5 types: *Yes/No question*, *Completion suggestion*, *Disjunctive question*, *Wh-question*, and *Phatic question*. The authors also categorized answers into 7 different types: *Positive answer*, *Negative answer*, *Feature answer*, *Phatic answer*, *Uncertainty answers*, *Unrelated Topic*, and *Deny the assumption*. They applied this annotation scheme to multiple languages (English, Spanish, and Dutch), and also offered an initial experiment for automating the annotation of question types in English dialogues. Cruz-Blandón et al. (2019) used 8 different hand-designed features and reported the classification results from both statistical machine learning algorithms (`Majority Baseline`: *acc.*=0.47, *F1*=0.31; `Decision Tree`: *acc.*=0.73, *F1*=0.58) and neural networks (`Bag-of-Words`: *acc.*=0.76, *F1*=0.44; `RNN`: *acc.*=0.54, *F1*=0.24).

3 A Taxonomy of Responses to Questions

As mentioned in the previous section, we deploy the corpus-based taxonomy proposed by (Ginzburg et al., 2019, 2022) in our study of automatic classification of response space of questions. They propose that the class of responses to a question q_1 can be classified into three main categories:

(1) a. Q(uestion)–specific: responses directly or indirectly about or subquestions of q_1;

b. MetaCommunicative: responses directly about or subquestions of a question defined in part from the *utterance* of q_1;

c. Evasion: responses directly about or subquestions of a question that is distinct from q_1 and arises from some other component of the context.

The first group is further classified as Direct Answers (DA) which constitute an answer to the initial question, and Indirect Answers (IND) through which one can infer an answer from its content, and also Dependent Questions (DP) where the answer to the initial q_1 depends on the answer to this query response. The second group is divided into Clarification Responses (CR) which inquire additional information to better understand the initial question, or to clarify some mis-presuppositions addressed in q_1. Acknowledgment (ACK) is the second class under the Metacommunicative group, which signals that the speaker heard and understood the q_1. The last group, Evasion responses, can be further categorized in to four response classes:

1. Ignore (IGNORE) (the utterance does not relate to the question, but to the situation. e.g., *A: So lock erm how would you spell sock? B: <laugh> smelly er smelly* (BNC));

2. Change the topic (CHT) (e.g., *A: Why couldn't they come on Friday? B: What you got me then?* (BNC));

3. Motive (MOTIV) *A: What's the matter? B: Why? (BNC)*;

4. Difficult to provide a response (DPR) (*A: When's the first consignment of Scottish tapes? B: Erm <pause> don't know.*).

The taxonomy is presented in Table 1.

Category	TAG
1. Direct answer	DA
2. Indirect answer	IND
3. Dependent question	DP
4. Clarification response	CR
5. Acknowledgment	ACK
5. The utterance does not relate to the question, but to the situation	IGNORE
6. Utterance signalizes that speaker does not want to answer, s(he) changes the topic, gives an evasive answer	CHT
8. Question about the motivation for the initial question	MOTIV
9. Difficult to provide an answer	DPR
10. Utterance that does not fit in any of the above	OTHER

Table 1: Taxonomy proposed by Ginzburg et al. (2022) and used in this paper

In the following section, we describe our data, annotation process, and also the inter-annotator agreement between annotators.

4 Response Space Annotation

Following the previous studies and the response space annotation guideline provided by Ginzburg et al. (2019, 2022), we annotated question-response pairs (QR-pairs) from different dialogue corpora. We manually annotated dialogues from the British National Corpus (BNC) (Burnard, 2007), Cornell-Movie (Danescu-Niculescu-Mizil and Lee, 2011), Basic Electricity and Electronic Corpus (BEE) collected from dialogue-based tutoring system (Rosé et al., 1999), and HCRC MapTask corpus (Anderson et al., 1991).

We manually annotated 3008 QR-pairs from the BNC corpus, 1172 QR-pairs from the Cornell-Movie, 293 QR-pairs from the HCRC MapTask, and 238 QR-pairs from the BEE corpus. This resulted in 4711 annotated QR-pairs in total. We have a rough estimate that more than 90% of the questions are responded to in the immediately following utterance. This is also in line with the statistics presented in (Purver et al., 2003b) that 94% of the Clarification Requests were answered in the immediately following utterance. Therefore, to facilitate the annotation and data processing for machine learning experiments, we only annotated QR-pairs where the response is the adjacent utterance of the corresponding question. In addition, we did not consider tag questions, such as, *It's too complicated, isn't it?* as a question. Finally, turns with missing text (the BNC's 'unclear') were eliminated from consideration, unless the remaining parts of the utterance provide sufficient information for understanding the meaning of the utterance.

To examine the annotation reliability, we double annotated three files from the BNC, and calculated the inter-annotator reliability based on the Cohen's κ (Carletta, 1996) and Krippendorff's α (Krippendorff, 2011) coefficients. The best inter-annotator agreement scores obtained are 0.8183 and 0.8186 for Cohen's κ and Krippendorff's α respectively. However, the lowest inter-annotator agreement scores are 0.7118 (Cohen's κ) and 0.7128 (Krippendorff's α).

Table 2 shows the distribution of the response space classes in our dataset. As can be observed from the table, the OTHER class is less than 1%, thus the coverage is more than 99%. What's more, the most frequent classes in our dataset are Direct Answers (64.83%), Indirect Answers (10.80%), Difficult to provide answer (5.20%), Change the topic (4.95%), and Clarification Responses (4.84%). The less frequent classes are DP (0.89%), MOTIV (0.30%), and ACK (3.12%).

The dataset used in this study is highly imbalanced, since the response class DA (64.83%) has significantly more samples than the others, as indicated in Table 2. Therefore, it is important to find a solution to overcome the classification difficulty caused by imbalanced data. In the following section, we introduce the baseline model obtained by the BERT pre-trained English language model (Devlin et al., 2018).

Category	**Total**	**Frequency%**
DA	3054	64.83%
IND	509	10.80%
DP	42	0.89%
CR	228	4.84%
ACK	147	3.12%
IGNORE	208	4.42%
CHT	233	4.95%
MOTIV	14	0.30%
DPR	245	5.20%
OTHER	31	0.66%
Total	**4711**	**100%**

Table 2: Overall distribution of response space classes in the dataset

5 Response Space Classification with BERT

To begin with, we set up an experiment with the pre-trained BERT language model, and examined the classification performance of such a large language model on the novel task of response space classification. First of all, we deleted all `OTHER` cases from our annotated dataset, which resulted in a total of 4680 annotated QR-pairs with 9 unique response classes. The distribution of the training, validation, and test sets are 60%, 20%, and 20% respectively. We add 2 special tokens <q> and <r> into BERT tokenizer's vocabulary, and the input of the BERT model is organized as {<q> *question* <r> *response*}.

We conducted two separate experiments: (1). with the full response space taxonomy of 9 unique classes; (2). with a coarser response space taxonomy of only 4 main classes, namely, Direct Answers, Indirect Answers, Clarification Responses, and Evasion. All classes which belong neither to Direct Answers, Indirect Answers, nor Clarification Responses were merged and classified as Evasion. We think that this is a more practical response space taxonomy in designing dialogue systems. In addition, we did not use any resampling techniques when classifying with the BERT language model, since BERT is already trained on a large amount of language data. Therefore, we are interested in seeing how it performs on this response space classification task with a skewed dataset.

Table 3 presents the classification results from the BERT language model on the full response space taxonomy. We use the classification results achieved by BERT model as the baseline for this study, and conduct several experiments to study whether we can obtain similar results as BERT by using classical machine learning algorithms trained with a set of carefully designed features.

As Table 3 shows, the baseline BERT model results in an average weighted f1-score of 0.70 and a macro f1-score of 0.40 on the full taxonomy. Besides, the BERT model achieved roc_auc scores of 0.87 and 0.86 respectively on the full and coarser taxonomy. This signals the very good performance of the BERT model on the response space classification task because they are very close to the perfect roc_auc score of 1.0. The best classified response class among others is the Direct Answers (f1-score: 0.85) as expected, since this is the easiest class to annotate for the human annotators according to the detailed human annotation report in Ginzburg et al. (2022). The next relatively well classified response classes are Clarification Responses (f1-score: 0.74), Acknowledgments (f1-score: 0.52), and DPR (f1-score: 0.59). This is also in line with the relatively higher inter-annotator agreement on these subsets of the full taxonomy, as presented in the previous response-space related literatures. However, the BERT model did not perform well on Indirect Answers, Dependent Questions, and other more evasive response classes, such as IGNORE, CHT, and MOTIV. The f1-scores are below 0.35 for these classes. Such low classification results were anticipated for response classes DP and MOTIV given the very low frequency of such responses in our dataset as shown in Table 2 (they comprise only 0.89% and 0.30% of the overall dataset). As for the response classes Indirect Answers, CHT, and IGNORE, even though their frequencies are higher than other non-major classes (10.80%, 4.95%, and 4.42% respectively), the classification results achieved by BERT language model are still very low (f1-score: 0.32, 0.33,

Classes	Precision	Recall	F1	Support
DA	0.81	0.88	0.85	593
IND	0.33	0.31	0.32	107
DP	0.10	0.20	0.13	5
CR	0.76	0.72	0.74	47
ACK	0.53	0.52	0.52	31
IGNORE	0.14	0.11	0.12	44
CHT	0.39	0.29	0.33	56
MOTIV	0.00	0.00	0.00	3
DPR	0.82	0.46	0.59	50
accuracy			0.70	936
macro avg.	0.43	0.39	0.40	936
weighted avg.	0.68	0.70	0.68	936
roc_auc_score				0.87
DA	0.77	0.95	0.85	595
IND	0.60	0.20	0.30	126
CR	0.70	0.63	0.67	41
Evasion	0.73	0.51	0.60	171
accuracy			0.75	933
macro avg.	0.70	0.57	0.60	933
weighted avg.	0.74	0.75	0.72	933
roc_auc_score				0.86

Table 3: Classification results of BERT language model on full and coarser response space taxonomy

and 0.12 respectively). This can be attributed to the fact that these response classes are intrinsically reliant on deep inference.

The bottom half of the Table 2 presents the classification results from BERT on the coarser taxonomy. The overall classification results improved in terms of the weighted average f1-score (0.75 vs. 0.70) on the coarser taxonomy. This was expected, since classifiers usually perform better on a coarser taxonomy. However, the f1-score on the classification results on Clarification Responses decreased from 0.74 to 0.67, and the Indirect Answers from 0.32 to 0.30. It can be observed that Indirect Answer is still the most difficult response class to be learned by the BERT language model. Finally, the model resulted in a f1-score of 0.60 on the classification of the Evasion response class, which is the new broader response class after merging all other response classes.

6 Classical Machine Learning Approach

In this section, we first introduce the set of carefully designed features for this response space classification task. Then, we present two groups of machine learning experiments: one with the full response space taxonomy, and the other with a coarser taxonomy.

6.1 Features

Similar to the approach used by Fernández et al. (2007), we also divided the features into three main groups: (i) `Response features`, which are related to properties of the response space; (ii) `Question features`, which are properties of the corresponding question; (iii) `Question-Response features`, which keep track of the features related to both question and response, and also similarities between the question and its corresponding response. All the semantic, syntactic, and lexical properties are extracted by using the Python natural language analysis package: Stanza Qi et al. (2020). Stanza is built with highly accurate neural network components that its neural network NLP pipeline can perform various NLP tasks, including tokenization, multi-word token expansion, lemmatization, POS and morphological tagging, dependency parsing, named entity recognition, and also the sentiment analysis of a natural language data. Table 4 presents the response space features and values used in this study.

`Response features` There are 12 different features related to the responses:

- `res_type`, `res_pers`, `res_number`, `res_tense`, `res_entities`, `res_sentiment`. The feature `res_type` has two values *question* and *proposition*, which are intended to capture the query responses and the propositional responses respectively. We encode the person information of the response with the feature `res_pers`. The feature `res_number` encodes the inflectional features of nouns in the response (*singular*, *plural*). `res_tense` records the time line in which the action in the response occurs (*present*, *future*, *past*). The feature `res_pers`, `res_number`, and `res_tense` use a value *empty* wherever the relevant lexical items are absent. Existence of name entities or proper nouns in the response is recorded with the feature `res_entities` (*yes*, *no*). The last feature `res_sentiment` is responsible for encoding the polarity of verbs, adjectives, adverbs, and nouns in the responses, with values *positive*, *negative*, and *neutral*.

- `rsp_aff` encodes the presence of affirmative word *yes* and *no*, we assign a value *empty* if there is no such word. `rsp_dntknow` has a value *yes* if there are phrases such as "I don't know", "dunno", "not sure", etc., and

Feature	Description	Values
res_type	query or propositional response	question, proposition
res_pers	person point of view in the response	1st, 2nd, 3rd, empty
res_number	inflectional feature of nouns	Sing, Plur, empty
res_tense	verb tense in the response	Pres, Fut, Past, empty
res_entities	presence of name entities	yes, no
res_sentiment	sentiment of the response	positive, negative, neutral
rsp_aff	presence of affirmative words	yes, no, empty
rsp_dntknow	presence of words indicating the absence of knowledge	yes, no
rsp_dprel_discourse	presence of "discoure" dependency	yes, no
rsp_dprel_reparandum	presence of "reparandum" dependency	yes, no
rsp_dprel_mwe	different multiword epresion dependency	compound, fixed, flat, empty
rsp_num_content	number of content words	integer
ques_type	wh-question or polar question	what,which...,polar
ques_pers	person point of view in the question	1st, 2nd, 3rd, empty
ques_number	inflectional feature of nouns	Sing, Plur, empty
ques_tense	verb tense in the question	Pres, Fut, Past, empty
ques_entities	presence of name entities	yes, no
ques_sentiment	sentiment of the question	positive, negative, neutral
ques_num_content	number of content words	integer
which_dem	presence of demonstative pronouns in responses utterance to *which* questions	yes, no
who_prs	presence of personal pronouns in responses utterance to *who* questions	yes, no
where_adp	presence of POS-tag "ADP-adposition" in responses to *where* questions	yes, no
wh_discorse	presence of "discourse" dependency in short responses to *wh* questions	yes, no
repeated_words	number of repeated words	integer
common_content_words	number of repeated common words	integer
pos_sequence	lenght of common POS sequence	integer

Table 4: Features of response space and values

no otherwise. rsp_deprel_discourse checks if there is a "discourse" dependency relation in the response utterance. rsp_deprel_reparandum looks for a "reparandum" dependency relation in the response utterance, which indicates disfluencies in the utterance. rsp_deprel_mwe encodes different dependency relations for multi-word expressions, and it has four values: "compound", "fixed", "flat", and "empty". Lastly, rsp_num_content presents the number of content words in the response utterance.

Question features We also use 7 different features to encode the properties of the corresponding questions, namely, ques_type, ques_pers, ques_number, ques_tense, ques_entities, ques_sentiment, and ques_num_content. The feature ques_type is used to differentiate the various types of wh- questions and polar questions. The other 6 features are used in a same way as the corresponding features in Response features described above.

Question-Response features
The last 7 features, repeated_word and pos_sequence, are the numerical features which encode features related to both question and response, and the similarities between the responses and their corresponding questions. The feature which_dem records the presence of demonstrative pronouns in a response utterance to a question with *which* ques_type. Similarly, the feature who_prs records the presence of personal pronouns in a response utterance to a question with *who* ques_type, and the feature where_adp records the presence of POS-tag "ADP-adposition" in a response utterance to a question with *where* ques_type. Besides, the feature wh_discourse indicates the presence of "discourse" dependency relation in short responses (less than or equal to two words) to any *wh-* questions. This feature aims to capture utterances such as "Aha", "Well", "Erm", "Mhm", etc, and they are usually classified as Acknowledgment to *wh-* questions. The feature repeated_word represents the number of repeated words between responses and questions; repeated_word shows the number of common content words in questions and responses; the feature pos_sequence records the length of the longest sequence of PoS tags common to responses and questions.

6.1.1 Experiment I: Classification with Over-sampling Method on Full Taxonomy

Data resampling is one of the most widely used methods for dealing with the imbalanced data problem. In this method, training instances are modified in order to produce a more balanced class distribution. One advantage of resampling techniques over other methods is that they are independent of the classifiers (López et al., 2013). The resampling techniques are mainly divided into two groups:

- Undersampling methods: this method generates a subset of the original dataset by deleting instances from the majority class. Random undersampling is a very simple non-heuristic method that randomly removes samples from the majority class. However, the drawback of random undersampling is that it may drop some potentially useful data that could be important for the classification.
- Oversampling methods: this method outputs a superset of the original dataset through replicating instances from minority classes. The non-heuristic simple random oversampling method balances the class distribution by randomly making exact copies of existing instances of the minority class. Therefore, the disadvantage of random oversampling is that it may cause overfitting.

In this study, we use the SVM-SMOTE over-sampling algorithms in the imbalanced-learn python package (Lemaître et al., 2017). We do not consider using the undersampling method because we do not have a huge amount of annotated data at this stage. SVM-SMOTE is a special variant of SMOTE algorithm (Chawla et al., 2003), which use an SVM algorithm to detect sample to use for generating new synthetic samples. This over-sampling algorithm resampled all response classes except from the majority class – Direct Answers.

For the classical machine learning task, we use the Support Vector Machine (SVM) classifier from the Scikit-learn library (Pedregosa et al., 2011; Buitinck et al., 2013). The Support Vector Classifier (SVC) internally always uses one-vs-one ('ovo') as a multi-class strategy to train models. However, we use the One-vs-Rest ('ovr') to return the decision function of shape (n_samples, n_classes) as all other classifiers. The One-vs-Rest ('ovr') method turns a multi-class classification into one binary classification problem per class. In addition, the balanced class-weights are used due to the imbalanced characteristics of our data sets.

Evaluation metrics: we report the classification results based on the precision, recall, and f1-score for each response class. Besides, we also show the average classification accuracy of all classes, macro average scores, and also the weighted average scores of precision, recall, and f1-score. Finally, we also present the average accuracy score resulting from 5-fold cross-validation, and also the Area Under the Receiver Operating Characteristic Curve (roc_auc_score) from prediction scores. Again, we use the One-vs-rest configuration to compute the AUC of each class against the rest. This 'ovr' method is sensitive to class imbalance, so it is more suitable for our imbalanced dataset.

Experimental results: Table 5 presents the classification performance of the SVM classifier on

Classes	Precision	Recall	F1	Support
DA	0.73	0.90	0.81	593
IND	0.38	0.19	0.25	107
DP	0.27	0.60	0.37	5
CR	0.67	0.77	0.71	47
ACK	0.33	0.58	0.42	31
IGNORE	0.33	0.02	0.04	44
CHT	0.38	0.09	0.14	56
MOTIV	0.00	0.00	0.00	3
DPR	0.85	0.34	0.49	50
accuracy			0.68	936
macro avg.	0.44	0.39	0.36	936
weighted avg.	0.64	0.68	0.63	936
SVM cv scores				0.85
roc_auc_score				0.79

Table 5: Classification results of SVM classifier on the full response space taxonomy with oversampling

Classes	Precision	Recall	F1	Support
DA	0.72	0.89	0.79	595
IND	0.42	0.04	0.07	126
CR	0.69	0.83	0.76	41
Evasion	0.43	0.34	0.38	171
accuracy			0.67	933
macro avg.	0.57	0.52	0.50	933
weighted avg.	0.62	0.67	0.62	933
SVM cv scores				0.82
roc_auc_score				0.79

Table 6: Classification results of SVM classifier on the coarser response space taxonomy with oversampling

the full response space taxonomy using the SVM-SMOTE oversampling method. As shown in the table, the SVM classifier achieved similar classification results as from the Bert model, in terms of weighted f1-score (0.63 – 0.68) and the macro f1-score (0.36 – 0.40) on the full response space taxonomy. The SVM classifier also performed well on some major response classes, such as Direct Answers (f1-score: 0.81) and Clarification Responses (f1-score: 0.71). However, despite the relatively high frequency of Indirect answers, both models did not perform well on identifying these response classes (f1-score: BERT - 0.32, SVM - 0.25). The overall trend of the classification results for other response class is also similar on both methods. Namely, the response classes such as IGNORE, MOTIV, and CHT are always the most difficult classes for both SVM classifier and BERT models. Moreover, both models can correctly capture nearly half the cases from Acknowledgments and DPR classes. Therefore, we argue that the feature sets designed to capture syntactic and lexical characteristics of responses and the corresponding questions are useful for recognizing some response classes, by merely using the most classical machine learning algorithms.

In addition, we also report the average accuracy from 5-fold cross validation during the training, and also the final roc_auc_score for the SVM classifier on the full taxonomy. The average accuracy from the cross-validation is 0.85%, and the roc_auc score is 0.79, which indicates a very good performance of our classifier. Since the roc_auc score is not affected by the imbalanced distribution of each class in the dataset, we think that roc_auc_score metric can better describe our model on response space classification task with a highly skewed dataset.

6.1.2 Experiment II: Classification with Over-sampling method on a Coarser Taxonomy

In the previous sections, we studied the automatic classification of 9 different response classes as described in Table 2. In this section, we are interested in studying the classification performance of the SVM classifier on a coarser response space taxonomy with only 4 distinct response classes, namely, Direct Answers, Indirect Answers, Clarification Responses, and Evasion.

As shown in Table 6, when classifying with a coarser taxonomy, the SVM classifier achieved a better macro average f1-score than on the full taxonomy (0.50 vs. 0.36). However, when compared to the results achieved by the BERT model (see Table 3) on the coarser taxonomy, the SVM model resulted in a lower weighted average f1-score (0.62 vs. 0.72) and macro average f1-score (0.50 vs. 0.60). The average accuracy for the 5-fold cross-validation while training is 0.82, and the roc_auc score is 0.79, which indicates a good performance of the SVM model. What is more, the overall trend of the classification results for each response class is similar to both the SVM model and the BERT model. Both models achieved similar high f1-scores for the Direct Answers, 0.79 and 0.85 respectively for the SVM and the BERT model. The second-highest performance score goes to the Clarification Responses on both models: f1-score is 0.76, and this is where our SVM model outperforms the BERT model (f1-score is 0.67 for Clarification Responses). However, the SVM model still failed to capture Indirect Answers and returned a 0.07 f1-score for this class. This is much worse than the f1-score of 0.30 achieved by the BERT model. Finally, the Evasion response class also caused many difficulties for both models, which resulted in f1-scores of 0.38 and 0.60 from the SVM and BERT model.

To conclude, regardless of the full or the coarser taxonomy, the DA response class is learned more easily by both pre-trained BERT language model and the classical machine learning algorithms. Whereas Indirect Answers, IGNORE, and MOTIV cause most difficulties for both models. In addition, the SVM model outperforms the BERT model on identifying Clarification Responses on this coarser taxonomy. Besides, the similar classification trend for each response class on both models suggests that the carefully designed feature sets are useful to capture the main response classes.

7 Conclusions and Future Work

We present a pilot study on the novel task of response space classification of questions in dialogue. We considered the classification results by the large scale pre-trained BERT language model with raw data (questions and responses) as baselines, and conducted experiments with more classical machine learning algorithms (the SVM classifier from the Scikit-learn library). We utilized 26 carefully designed syntactic and lexical features on the SVM classifier, which aim to capture characteristics of responses and question. Since the class distribution in our datasets is highly imbalanced, we first deployed an over-resampling methods to mitigate the imbalanced data problem. Then, we conducted two groups of experiments respectively on both BERT and SVM models: (1) with a fine-grained full response space taxonomy with 9 unique response classes, and (2) with a coarser taxonomy with only 4 main response classes. Finally, we compared the classification results from both models and offered detailed discussions regarding the differences and similarities observed from two models.

The main contributions of this study are threefold: (1) To our knowledge, this is the first study on the automatic classification of response space of questions in dialogue. Such a classification task is of great importance in the design of dialogue systems, particularly systems that can be easily adaptable across domains. (2) We designed 26 different features which help the classical machine learning algorithms to correctly identify different response classes; (3) We provided detailed discussion of the learnability of various response classes by the pre-trained language model and the classical SVM classifier, and observed that the learnability trend is closely in line with that achieved by the human annotators in previous work.

However, we also acknowledge the limitations of the current study and have some initial thoughts for future studies. Firstly, we hope to scale-up the current feature sets used for the SVM model by designing more useful features in terms of syntactic, semantic, and lexical relationships between questions and responses. Secondly, since dialogues are highly context-dependent interactions, we also want to conduct experiments by adding features pertaining to such aspects to the feature set, e.g., the number of common words between previous utterances and questions/responses, the length of the previous utterances etc. Thirdly, a detailed analysis of which features are more informative and which are redundant can also be very useful for the classification task. Lastly, more carefully created features targeting Indirect Answers are necessary to correctly classify this highly inference-based response class.

Acknowledgments

We acknowledge the support by a public grant overseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (reference: ANR-10-LABX-0083). It contributes to the IdEx Université Paris Cité ANR-18-IDEX-0001. We also acknowledge that the second author is supported by the French government under the management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). We thank the SemDial reviewers for very helpful comments.

References

Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth H. Boyle, Gwyneth M. Doherty, Simon C. Garrod, Stephen D. Isard, Jacqueline C. Kowtko, Jan M. McAllister, Jim Miller, Catherine F. Sotillo, Henry S. Thompson, and Regina Weinert. 1991. The HCRC Map Task Corpus. *Language and Speech*, 34(4):351–366.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD*

Workshop: Languages for Data Mining and Machine Learning, pages 108–122.

Lou Burnard, editor. 2007. *Reference guide for the British National Corpus (XML Edition)*. Oxford University Computing Services on behalf of the BNC Consortium. Acess 20.03.2017.

Jean Carletta. 1996. Assessing agreement on classification task: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. 2003. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer.

William W Cohen and Yoram Singer. 1999. A simple, fast, and effective rule learner. *AAAI/IAAI*, 99(335-342):3.

Maria-Andrea Cruz-Blandón, Gosse Minnema, Aria Nourbakhsh, Maria Boritchev, and Maxime Amblard. 2019. Toward dialogue modeling: A semantic annotation scheme for questions and answers. *arXiv preprint arXiv:1908.09921*.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2003. Timbl: Tilburg memory based learner, v. 5.0. Technical report, Reference Guide. Technical Report ILK-0310, University of Tilburg.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd workshop on cognitive modeling and computational linguistics*, pages 76–87. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paolo Dragone and Pierre Lison. 2015. An active learning approach to the classification of non-sentential utterances. In *Proceedings of the Second Italian Conference on Computational Linguistics*, pages 115–119.

Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2007. Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics*, 33(3):397–427.

Jonathan Ginzburg, Zulipiye Yusupujiang, Chuyuan Li, Kexin Ren, Aleksandra Kucharska, and Paweł Łupkowski. 2022. Characterizing the response space of questions: data and theory. *Dialogue and Discourse (accepted)*. `https://drive.google.com/file/d/1AieL7JERQhJnTPlbgn1P_YPDaLP8gGJl/view`.

Jonathan Ginzburg, Zulipiye Yusupujiang, Chuyuan Li, Kexin Ren, and Paweł Łupkowski. 2019. Characterizing the response space of questions: a corpus study for english and polish. In *Proceedings of the 20th annual SIGdial meeting on discourse and dialogue*, pages 320–330.

Jacqueline C. Kowtko and Patti J. Price. 1989. Data collection and analysis in the air travel planning domain. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '89, pages 119–125, Stroudsburg, PA, USA. Association for Computational Linguistics.

Klaus Krippendorff. 2011. Agreement and information in the reliability of coding. *Communication Methods and Measures*, 5(2):93–112.

Staffan Larsson and Alexander Berman. 2016. Domain-specific and general syntax and semantics in the talkamatic dialogue manager. *Empirical Issues in Syntax and Semantics*, 11:91–110.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250:113–141.

Paweł Łupkowski and Jonathan Ginzburg. 2013. A corpus-based taxonomy of question responses. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 354–361, Potsdam, Germany. Association for Computational Linguistics.

Paweł Łupkowski and Jonathan Ginzburg. 2016. Query responses. *Journal of Language Modelling*, 4(2):245–293.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk*, third edition. Lawrence Erlbaum Associates, Mahwah, NJ.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew Purver. 2006. Clarie: Handling clarification requests in a dialogue system. *Research on Language and Computation*, 4(2):259–288.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2003a. On the means for clarification in dialogue. In *Current and new directions in discourse and dialogue*, pages 235–255. Springer.

Matthew Purver, Patrick Healey, James King, Jonathan Ginzburg, and Greg J Mills. 2003b. Answering clarification questions. In *Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue*, pages 23–33.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Kepa Joseba Rodríguez and David Schlangen. 2004. Form, intonation and function of clarification requests in german task-oriented spoken dialogues. In *Proceedings of Catalog (the 8th workshop on the semantics and pragmatics of dialogue; SemDial04)*. Citeseer.

Carolyn P. Rosé, Barbara Di Eugenio, and Johanna D. Moore. 1999. A dialogue-based tutoring system for basic electricity and electronics. In Susanne P. Lajoie and Martial Vivet, editors, *Artificial intelligence in education*, pages 759–761. IOS, Amsterdam.

Ian H Witten and Eibe Frank. 2002. Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, 31(1):76–77.

L Zhang. 2007. Maximum entropy modeling toolkit for python and c++ (online). *http://homepages. inf. ed. ac. uk/s0450736/maxent_toolkit. html*.